Robust Spatial-Domain Watermarking Methods Based on a Weighting Table with Fine-Tune Technique

Mu-San Chung, Kai-Hsiang Chang and Shen-Fu Hsiao
Dept. of Computer Science and Engineering, National Sun Yat-Sen University, Kaohsiung, Taiwan, R.O.C.

Abstract

Spatial-domain digital watermarking methods are generally considered as having poor performance after geometric distortion (such as cropping and scaling), common signal processing (such as JPEG and filtering), or subterfuge attacks (such as changing the least-significant bits). On the other hand, frequency-domain watermark techniques, in addition to their high computation complexity, usually require original image data during extraction. The methods, without referring to the original image during watermark extraction, are based on a predefined weighting table to record the magnitude difference of image pixel values and a novel fine-tune process to gradually change pixel values in order to reduce the impact on image quality. The performance can be further improved by properly selecting reference pixels for each partitioned 8×8 *block and determining the reliability metrics for proper weighting during watermark extraction. Experimental results show promising robustness performance compared to other spatial-domain and frequency-domain methods.*

1. INTRODUCTION

Due to the fast development of multimedia technologies and the wide distribution of image data over internet, information security is becoming an important issue. For example, encryption, digital signature and digital watermark are three popular methods to prevent a wide variety of multimedia products from illegal copy or subterfuge attacks. In particular, digital watermarking recently draws a lot of attention since it hides desirable information in transmitted audio, image and video data files without affecting much the data quality [1][2]. In general there are four criteria to judge the performance of a watermarking method: invisibility, robustness, security, and capacity [2][5]. A watermarked image should be perceptually undetectable and peak signal-to-noise ratio (PSNR) is a popular measure of the perceptual transparency. The watermark information should be difficult to remove after common signal processing (JPEG, filtering), common geometric distortions (rotation, scaling, cropping, translation), and subterfuge attacks (collusion, forgery). In many applications, the embedding procedure must be also secure in that an unauthorized user can not detect the presence of the hidden information. It is also desirable to add as much watermark information as possible. Another important issue of watermarking is the recovery of data without access to original data. In some applications, such as copy tracking and copyright protection, the data-extraction algorithms may use the original signal to decode the embedded data. However, most multimedia applications do not have the ability to access the original data and thus the recovery of embedded information without original signal becomes an important consideration during these applications.

There have been a lot of papers discussing various watermarking methods. In general, the watermarking methods can be divided into two categories: spatial-domain and frequency-domain watermarking [3][4][6][7][8]. Spatial-domain methods can add more watermark information but they usually are not robust to common signal processing and geometric distortion. Most of recently proposed watermarking schemes embedded watermark bits into transformed image coefficients [1][8]. Although these computation-intensive watermarking methods have much better robustness performance, they usually require the access of the original signal during extraction. In this paper, we will present two novel spatial-domain watermarking methods using a weighting table to record the change of pixel values and then fine tune the pixel values according to the watermark bits to be added. Our proposed method has better overall performance in common signal processing and does not require original data to extract the embedded watermark information.

2. EMBEDDING METHODS

Spatial domain watermark embedding methods are usually vulnerable to common signal processing, as will be seen later in Sec. 4 of experimental results. The major reason for such a poor performance is that the least-significant bits (LSBs) of a pixel value are changed during common signal processing such as JPEG encoding/decoding or filtering. According to our survey, most spatial-domain watermark methods affect only the least three significant bits in order to preserve the quality of the original image. The change of the fourth LSB for each pixel value usually has significant influence on image quality that is perceptible to human visual system.

We exploit the above observation by constructing a weight table for the transform of pixel values, as shown in Tab. 1. Each pixel value is mapped to a positive or negative weighting value. Only the four LSB bits are considered during the mapping process, with the fourth LSB determining the sign of the weighting value. The difference of two weighting values for two pixels indicates the probability of one pixel being changed to the other if the magnitude difference of the two pixels is within 4. Smaller difference of two weighting values

Pixel value at low bits	Pixel weight	Pixel value at low bits	Pixel weight
0000	-0.1	1000	0.1
0001	-0.3	1001	0.3
0010	-0.7	1010	0.7
0011	-0.9	1011	0.9
0100	-1	1100	1
0101	-0.9	1101	0.9
0110	-0.7	1110	0.7
0111	-0.3	1111	0.3

Table 1: The weighting table

corresponds to higher probability. For example, a pixel value of 11000000 with four LSBs of 0000 is mapped to the weighting of –0.1 while another pixel value 11000001 with the same four MSBs but with different four LSBs of 0001 is mapped to the weighting of -0.3. The difference of the magnitudes of these two pixels is only 1, which means that it is very likely that the first pixel value of 11000000 is changed to the second pixel value of 11000001 under some kind of processing. The idea of embedding watermark information in this paper is to gradually change the pixel value until the corresponding weighting achieves a predefined threshold value. The process of changing the pixel values is called *fine-tune* process whose objective is to have as small pixel magnitude change as possible while still maintaining the robustness of the watermarking scheme.

2.1 Basic Watermark embedding scheme

In order to increase the robustness, we consider adding one watermark bit for a group of pixels, say four pixels for each watermark bit as will be shown shortly. Assume that the size of the original gray-level image is $N \times N$ and the watermark bits are from a binary image of $M \times M$ with pixel values $\in \{1, -1\}$. The watermark embedding procedure is listed below and illustrated in Fig. 1.

E-(1): Divide the original image into four regions (A,B, C, D) of the same size.

E-(2): Partition each region into blocks of size 8×8.

E-(3): Calculate the variance of all the 8×8 blocks and select the $M^2/64$ blocks with the largest variances for watermark embedding. Number the blocks according to the order of their variance. If we want to reduce the computation complexity, this step of block variance calculation can be skipped by selecting randomly $M^2/64$ blocks in each region instead. However, embedding watermark bits into image blocks of large variance will reduce the influence on image quality since human visual system is less sensitive to the pixel value changes of image blocks with large variance.

E-(4): Denote the pixel value at the *j*-th position of the *i*-th 8×8 block in region *A, B, C, D* as $A_{i,j}, B_{i,j}, C_{i,j}, D_{i,j}$ respectively where $1 \le i \le M^2/64, 1 \le j \le 64$.

E-(5): Map a group of four pixels $A_{i,j}, B_{i,j}, C_{i,j}, D_{i,j}$ into their corresponding weightings $A'_{i,j}, B'_{i,j}, C'_{i,j}, D'_{i,j}$ using Tab. 1.

E-(6): Based on the *k*-th binary watermark bit $W_k \in \{1, -1\}$, calculate if the following inequality (1) is satisfied

$$(A'_{i,j}, +B'_{i,j} + C'_{i,j}, +D'_{i,j}) \times W_k \ge T_1 \qquad (1)$$

where T_1 is the prescribed threshold that is a function of the number of pixels in each group. Note that the selection of this threshold affects the image quality and the robustness performance of the watermarking scheme. In our experiment, T_1 is set to be 25 for group size of 4. If inequality (1) is satisfied, the watermark bit W_k has been already added to the group of four pixels $A_{i,j}, B_{i,j}, C_{i,j}, D_{i,j}$. Then continue the embedding steps E-(5) and E-(6) for another group of four pixels. If not, proceed to the following fine-tune process.

E-(7): Based on the sign of the watermark bit W_k, change the magnitudes of $A_{i,j}, B_{i,j}, C_{i,j}, D_{i,j}$ to their neighboring pixels $\tilde{A}_{i,j}, \tilde{B}_{i,j}, \tilde{C}_{i,j}, \tilde{D}_{i,j}$ with the weightings closer to the value W_k. Then continue steps E-(5) and E-(6).

The following exemplifies our watermark embedding procedure. Assume that we want to add a watermark bit W_k = -1 into the group of four pixels with values

$$A_{i,j} = 10111111,\ B_{i,j} = 11000000,$$
$$C_{i,j} = 00100111,\ D_{i,j} = 00110011$$

According to the pixel weighing in Tab. 1, we have the corresponding weightings

$$A'_{i,j} = 0.3, \quad B'_{i,j} = -0.1, \quad C'_{i,j} = -0.3, \quad D'_{i,j} = -0.9$$

The sum of the above four weightings is –1 and inequality (1) is not satisfied. Thus, we change the four pixels to their neighboring pixels with weightings closer to W_k = -1, i.e.,

$$\tilde{A}_{i,j} = A_{i,j} + 1 = 11000000, \quad \tilde{B}_{i,j} = B_{i,j} + 1 = 11000001,$$
$$\tilde{C}_{i,j} = C_{i,j} - 1 = 00100110, \quad \tilde{D}_{i,j} = D_{i,j} + 1 = 00110100$$

The corresponding weightings of the above fine-tuned pixels are

$$\tilde{A}'_{i,j} = -0.1, \quad \tilde{B}'_{i,j} = -0.3, \quad \tilde{C}'_{i,j} = -0.7, \quad \tilde{D}'_{i,j} = -1$$

The sum of the above four weighting values does not satisfy inequality (1) either, and thus we continue another fine-tune process by replacing $\tilde{A}_{i,j}, \tilde{B}_{i,j}, \tilde{C}_{i,j}, \tilde{D}_{i,j}$ with their neighboring pixels.

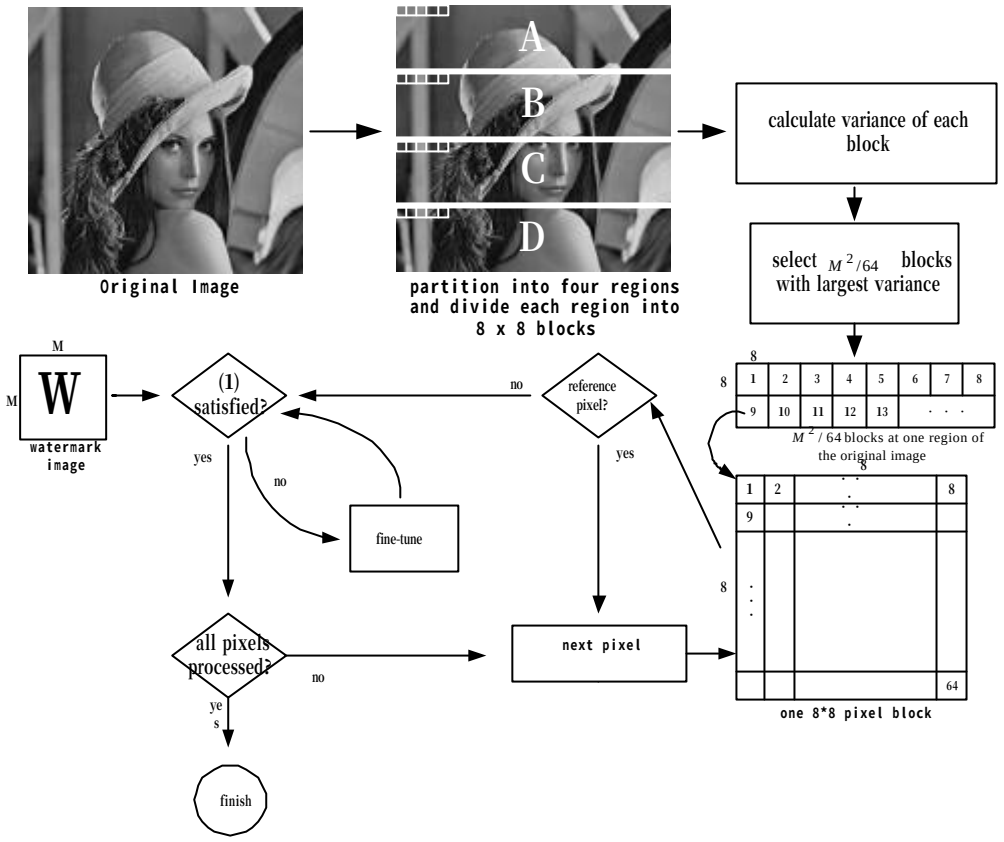

Fig. 1: The watermark embedding procedure.

$\hat{A}_{i,j} = \tilde{A}_{i,j} + 1 = 11000001$, $\hat{B}_{i,j} = \tilde{B}_{i,j} + 1 = 11000010$,
$\hat{C}_{i,j} = \tilde{C}_{i,j} - 1 = 00100101$, $\hat{D}_{i,j} = \tilde{D}_{i,j} = 00110100$

The corresponding weightings are

$$\hat{A}'_{i,j} = -0.3, \quad \hat{B}'_{i,j} = -0.7, \quad \hat{C}'_{i,j} = -0.9, \quad \hat{D}'_{i,j} = -1$$

The above weighting satisfies inequality (1) and thus the fine-tune process terminates.

2.2 Improved Watermark Method with Reference Pixels

We can increase the robustness and security of the above watermarking method by dividing the pixels in each 8 x 8 block into two categories: reference pixels and embedding pixels. For example, the pixels in the 8 x 8 block of Fig. 2 are divided into 48 embedding pixels and 16 reference pixels where the reference pixels are located in three diagonals. The watermark information is added into the embedding pixels while the reference pixels are used to detect the *reliability* of this 8 x 8 block. Indeed, after some geometric distortion, pixel values in some blocks might be changed significantly while pixel values in other blocks have small changes. We can use the reference pixels to calculate the degree of changes under a specific signal processing. In this case, each 8 x 8 block should be assigned a factor indicating reliability (fidelity) of this block. Using the same mapping table of Tab. 1, we obtain the weightings of the 16 reference pixels in the 8 x 8 block of Fig. 2 and denote them as $R'_k, k = 1,2,\Lambda\ ,16$. Let

$$P = \sum_{k=1}^{16} R'_k (-1)^k \qquad (2)$$

The same fine-tune method as in Sec. 2.1 is used for the 16 reference pixels so that inequality $P \geq T_2$ is satisfied. The threshold value T_2, affecting the reliability, is selected to be 10 in our experiment. The watermark embedding method for other embedding pixels is the same as discussed before in Sec. 2.1. Later in Sec. 4, we will see that such a watermark embedding method with reference pixels has better performance. Since only 48, instead of 64, watermark bits are added for each 8 x 8 block, the number of blocks selected for watermark embedding in each region should be $M^2/48$

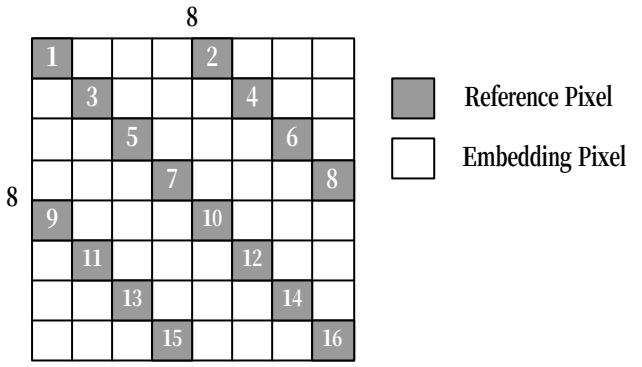

Fig. 2: Embedding pixels and reference pixels in an 8 x 8 block.

instead of $M^2/64$. Although the positions of the reference pixels in each block are fixed in Fig. 2, it is possible to arbitrarily select the positions of these reference pixels in order to increase the security of the watermark method. For example, the location of the reference pixels may come from a pseudo random sequence generator.

3. WATERMARK EXTRACTION AND DETECTION

The watermark decoding process contains two stages. The first stage is to extract watermark sequence from the received image, which might undergo any kind of signal processing or attack. The second stage is to detect whether the received image contains the desired watermark based on the extracted sequence and the original watermark sequence. We will discuss these two stages in the following.

3.1 Extraction of Watermark Sequence

Our proposed watermark extraction method does not require the access of the original image data. The extraction process is quite simple. Assuming that $A'_{i,j}, B''_{i,j}, C''_{i,j}, D''_{i,j}$ are the corresponding weighting values of the four pixels from the same position of the four regions in the received image, we can determine the corresponding watermark bit W'_k as follows:

$$\text{If } (A''_{i,j} + B''_{i,j} + C'_{i,j} + D''_{i,j}) \geq 0, \quad \text{then} \quad W'_k = 1 \quad \text{else} \quad W'_k = 0 \tag{3}$$

In other words, the sign of the summation of the corresponding four weighting values determines the embedded watermark bit.

If the reference-pixel method proposed in Sec. 2 is adopted to enhance the robustness, the above watermark extraction is modified as follows:

$$\text{If } (P_i^{(a)} A'_{i,j} + P_i^{(b)} B'_{i,j} + P_i^{(c)} C''_{i,j} + P_i^{(d)} D'_{i,j}) \geq 0 \quad \text{then} \quad W'_k = 1; \quad \text{else} \quad W'_k = 0 \tag{4}$$

$P_i^{(a)}, P_i^{(b)}, P_i^{(c)}, P_i^{(d)}$ are respectively the reliability metrics of the four i-th 8×8 blocks where the four pixels with weightings $A''_{i,j}, B''_{i,j}, C''_{i,j}, D''_{i,j}$ are located. The four reliability metrics, calculated using Eqn. (2), are used as the scaling factors to determine the confidence of the four weightings $A''_{i,j}, B''_{i,j}, C''_{i,j}, D''_{i,j}$ if the pixel values in some part of the image encounter significant changes due to geometric distortion, common signal processing, or other subterfuge attacks. Recall that in the watermark embedding process mentioned in Sec. 2, the reliability metric of an 8×8 block is fine-tuned to be greater than or equal to a threshold value T_2 ($T_{2=10}$ in our experiments). Thus, if any one of $P_i^{(a)}, P_i^{(b)}, P_i^{(c)}, P_i^{(d)}$ is greater than 10, we limit it to 10 while if some of $P_i^{(a)}, P_i^{(b)}, P_i^{(c)}, P_i^{(d)}$ are smaller than 1 (due to the serious attacks), we assign the value of 1 for them. If the four values 8×8 are in between, i.e., $1 \leqq P_i^{(a)}, P_i^{(b)}, P_i^{(c)}, P_i^{(d)} \leqq 10$, their values are used as the scaling factors for inequality (4) during the watermark extraction. Inequality (3) is in fact a special case when the scaling factors $P_i^{(a)}, P_i^{(b)}, P_i^{(c)}, P_i^{(d)}$ of the i-th block are all one. Shortly, we will see that the watermark scheme with reference-point has better performance.

3.2 Watermark Detection

After extracting watermark bit sequence W'_k, we calculate the normalized correlation of W'_k with the original pseudo random sequence W_k in order to determine whether the extracted sequence is the correct watermark bit sequence. The normalized correlation of the two sequences are defined as

$$Corr(W, W') = \frac{IP(W, W')}{\sqrt{IP(W, W) \times IP(W', W')}}$$

where $IP(A, B)$ is the inner product o f two sequences A_k, B_k. If the normalized correlation of the two sequences is larger than a threshold value, i.e.,

$$Corr(W, W') \geq T_3 \tag{5}$$

the extracted watermark sequence is determined to be the same as the original embedded sequence. The selection of the threshold value T_3 affects the false detection rate or false pass rate. In the following, we give a systematic method to choose this threshold value.

(a) original Lena (b) watermark Lena-I (PSNR= 44.7) (c) watermark Lena-II (PSNR= 43.4)

(d) original baboon (e) watermarked baboon-I (PSNR=46.1) (f) watermarked baboon-II (PSNR= 46)

Fig. 3: The original images of Lena (a) and baboon (d) and those after embedding watermark bits using our proposed methods without (b)(e) and with (c)(f) reference pixels.

According to the central limit theorem, the normalized correlation $X = Corr(W, W')$ of the extracted watermark sequence and all the other pseudo random sequences can be approximated as a random variable with Gaussian distribution. Thus, $Z = \frac{X - m}{s}$ with sample mean m and sample variance s^2 is a normalized Gaussian distribution $Q(z)$ with zero mean and standard deviation of one

$$Q(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2p}} e^{-\frac{x^2}{2}} dx$$

. Let the threshold is selected as $T_3 = as$, the probability of false watermark detection is

$$\text{Prob}\{X > T_3\} \le Q(a). \quad (6)$$

In other words, the probability of detecting the watermark sequence for non-watermark images is less than $Q(a)$. Thus, we can select the threshold value T_2 depending on this probability which might vary from applications to applications.

4. EXPERIMENT RESULTS

The test images we use in this experiment are Lena and baboon of size 256 x 256 with gray levels of 8 bits for each pixel. The embedded binary watermark bit sequence with length of 32*32 is the 200th sequence among the 1000 pseudo-randomly generated sequences. For convenience, our proposed watermark methods without and with reference pixels are noted respectively as method-I and method-II. We first examine the invisibility of our proposed watermark methods by calculating the peak-signal-to-noise-ratio (PSNR) for the two watermark images. It is almost imperceptible as can be observed Fig. 3.

Next, we compare the performance of our proposed watermark methods with other approaches under a variety of signal processing or attacks as shown in Tab. 2 for Lena image. Two types of geometric distortion (cropping and dilation), two types of common signal processing (JPEG and equalization) and the subterfuge attack of the last three LSB bits are considered. The values in Tab. 2 are the normalized correlation of the extracted watermark sequence with the original sequence. Based on the analysis in Sec. 3, the threshold of the correlation is set to be 0.1 with the probability of false detection smaller than 10^{-4}. In other words, watermark sequences with correlation smaller than 0.1 is determined to be undetectable since they are not significantly larger than

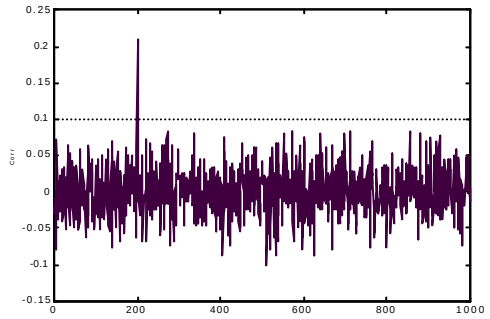

Fig. 4: The normalized correlation of the extracted watermark sequence with all the 1000 pseudo random sequences (our proposed method under JPEG processing).

the correlation with other no-watermark sequence. Since the embedded watermark bits are from the 200th sequence, the correlation of the extracted watermark sequences with the 200th sequence should be much larger than the correlation with all the other 999 random sequences, as indicated in Fig. 4. From Tab. 2, we have the following observations:

(1) Most of the digital watermark methods have high PSNR (greater than 40) except for the method proposed by Cox [1]. As can be seen from Fig. 3, images with PSNR greater than 40 are almost indistinguishable with the original images.

(2) Most of the frequency-domain methods require the original image during watermark extraction. Thus, their applications are limited and a large memory space is needed to store the original images before watermark extraction.

(3) Compared with frequency-domain approaches, the spatial-domain watermarking methods usually have better performance under geometric distortion but have poor performance under common signal processing and subterfuge attacks. For example, the normalized correlation values for the listed two frequency-domain methods under 50% cropping are around 0.3-0.4 compared to those of 0.7-0.8 using spatial-domain methods. On the other hand, the correlation values of the frequency-domain methods under JPEG processing with 1.8 bit-per-pixel (bpp) are 0.7-0.8, much higher than those using spatial-domain methods.

(4) Most spatial-domain methods are vulnerable to common signal processing such as JPEG and filtering. For example, the correlation values using the LSB method and the method proposed in [8] are smaller than the threshold value of 0.1. Hence, the embedded watermark information is lost. However, our proposed methods are robust to this kind of signal processing or filtering.

(5) Our proposed method with reference pixels have better robustness performance than without using reference points at the cost of slightly lower PSNR (but still greater than 40). For example, the correlation value is increased from 0.43 to 0.85 for the watermark image with dilation.

(6) Our proposed spatial-domain watermarking methods have better overall performance and they do not need the original image during watermark extraction. Furthermore, the methods are simple and easy to realize using either hardware or software.

We also tested three benchmark images (Lena, baboon and F-16) using our proposed watermark methods with and with reference pixels. The experimental results are shown in Tab. 3 where ours-I is the method without reference pixels while ours-II is that with reference pixels. The watermark embedding method with reference pixels is in general has better performance.

5. CONCLUSIONS

We proposed two spatial-domain watermarking methods based on a pixel-weighting table and the fine-tune process. The methods, although simple, have better robustness performance compared to other spatial-domain or frequency-domain methods. Furthermore, our methods do not require the original image during watermark extraction, make them favorable for most watermark applications.

References

[1] I. J. Cox, J. Kilian, T. Leighton and T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia", *IEEE Trans. On Image Processing*, Vol. 6, No12, pp.1673-1687, Dec. 1997.

[2] M. D. Swanson, M. Kobayashi, and A. H. Tewfik, "Multimedia Data-Embedding and Watermarking Technologies", *Proc. of IEEE*, Vol. 86, No.6, pp.1064-1087, June 1998.

[3] C.-T. Hsu and J.-L. Wu, "Hidden Digital Watermark in Image", IEEE Transactions on Image Processing, Volume: 8 1, pp 58 –68, Jan. 1999.

[4] W. Tang and Y. Aoki, "A DCT-based Coding of Images in Watermarking", Information, Communications and Signal Processing., Proc. ICICS, Volume: 1 , pp 510 -512, 1997.

[5] I. J. Cox, Jean-Paul M. G. Linnartz, "Some General Methods for Tampering with Watermarks", , IEEE Journal on Selected Areas in Communications, Volume: 16 4 , pp 587 -593 , May 1998.

[6] R. B. Wolfgang and E. J. Delp, "A Watermark for Digital Image", Proc. IEEE Intl. Conf. on Image Processing, Volume: 3 ,pp 219 -222, 1996 .

[7] M. Wu, and B. Liu, "Watermarking for image authentication", Proc IEEE Intl. Conf. on Image Processing, vol. 2, pp. 437-441, 1998.

[8] W. Zhu, and Z. Xiong and Y. Q. Zhang, "Multi-resolution watermarking for images and video", IEEE Transactions on Circuits and Systems for Video Technology, vol. 9, no. 4, pp. 545-550, Jun. 1999.

Digital watermarking methods		Original image needed?	PSNR (db)	Geometric distortion		signal processing		Subterfuge attack
				cropping (50%)	dilation	JPEG (1.8bpp)	Equalization	Three LSB bits changed
Spatial-domain	Ours-I	No	44.7	0.70	0.43	0.35	0.30	0.85
	Ours-II	No	43. 4	0.75	0.85	0.50	0.86	0.95
	LSB	No	52.0	0.74	0.68	0.01	0.02	0.07
	Wu [7]	No	55.0	0.62	0.46	0.01	0.03	0.02
Freq.-domain	Cox [1]	Yes	35.0	0.35	0.33	0.81	0.50	0.72
	Zhu [8]	Yes	42.0	0.37	0.27	0.75	0.43	0.74

Tab. 2: Comparison of robustness of different watermarking methods under a wide variety of signal processing.

Three test images		PSNR	Geometric distortion		signal processing	
			cropping	dilation	JPEG	Equalization
Lena	Ours-I	44.7	0.70	0.43	0.35	0.30
	Ours-II	43.4	0.75	0.85	0.5	0.86
baboon	Ours-I	46.1	0.78	0.43	0.20	0.10
	Ours-II	46	0.78	0.74	0.10	0.10
F16	Ours-I	44.8	0.86	0.21	0.32	0.18
	Ours-II	43	0.69	0.94	0.49	0.90

Tab. 3: Performance comparison of our proposed watermark methods for different benchmark images.